# Adaptive Systems Using Generative AI

2024-1-9


**Authors:**

Dhakshinamoorthy Renganathan
Tata Consultancy Services - IoT DE
*dhaks.r@tcs.com*

Suresh Kaluvan
Tata Consultancy Services - IoT DE
*suresh.kaluvan@tcs.com*

## CONTENTS



## FIGURES

# 1 OVERVIEW

Generative AI technology is making big waves. It's not just about using data to make decisions anymore; it's about generating new data from scratch. Generative AI opens multiple possibilities and use cases can be identified for each stage in the value chain of any industry.

This paper explores the possibility of utilizing Generative AI to build adaptive systems for industrial context. The article draws lessons from industries such as telecom, mobile phones and automotive on how software centric concepts have helped impart resilience, adaptiveness and support to new business models. We look at specific examples of applications of Generative AI to support adaption in robotics and multi-agent systems. We then explore how Generative AI, along with digital twins, can be used to build adaptive systems.

# 2 INTRODUCTION

Adaptive systems can be defined as systems that can adjust their behavior and/or structure in response to changes in environment or its own components. Examples of adaptive systems can best be taken from biological systems such as cells and micro-organisms. Over millions of years, cells and micro-organisms have survived harsh changes in the environment and their own structures by adapting and being resilient. For our purpose, we will call these self-adaptive systems.

We will define adaptive systems as systems that can adjust their behavior and/or structure in response to changes in environment or components based on instructions from an operator (employee of the enterprise) or a customer. These human-in-middle adaptive systems still need to adhere to the secure governance policies established by the enterprise to ensure the adaptive systems operate within the boundaries of intended purpose.

Studying behavior of complex biological structures such as an ant colony[1], self-adaptive systems have the following characteristics[2]:

- The system is comprised of multiple heterogeneous agents. Each agent has well defined objective and behavior. Each agent learns and follows the objective for the agent.
- The agents interact. The interactions are defined by purpose. The agents exhibit adaptive behavior and learn/adapt from experience or influence.
- Exhibit behavior of emergence where the whole is more than sum of parts and gives nonlinear benefits.

[1] *https://hbr.org/2011/09/embracing-complexity*

[2] *https://www.sciencedirect.com/science/article/pii/B9780128037263000079*

In this article, we will look at learnings from specific industries that have leveraged these adaptive characteristics. We will also look at how Generative AI can enable enterprises to absorb adaptive behavior in their systems and products.

## 2.1 Purpose

The purpose of this article is to inform and offer guidance to a wide range of stakeholders encompassing product designers, design engineers, manufacturers, and architects in implementing adaptive industrial systems/products. The objective is to build systems/products that are adaptive, resilient and provide customer appreciation. Products that can have features added based on consumer inputs will disrupt and establish new business models.

## 2.2 Scope

In this article, we focus on the development of human-in-middle adaptive systems using Generative AI. We explore the tools and frameworks available to support this endeavor. We do not cover the ethics and governance aspects as it will require a separate article.

## 2.3 Structure

The article is organized as follows:

- Chapter 3 – Motivation for Adaptive Systems
- Chapter 4 – Lessons from Industries
- Chapter 5 – Generative AI Applied for Adaptive Systems (i.e. Robots)
- Chapter 6 – Digital Twins: The Pathway to Adaptive Systems
- Chapter 7 – Summary

## 2.4 Audience

The article provides new insights to CXOs, product designers, design engineers, manufacturers, and architects in designing and implementing adaptive industrial systems/products.

## 2.5 Use

The article explores and provides insights on developing adaptive industrial products/systems.

## 2.6 Terms and Definitions

The following are terms and definitions that are key to understanding this document:

- **SDN –** Software Defined Network uses programmable software controllers that allow the physical network to be controlled by software.
- **SDV –** Software Defined Vehicles are vehicles whose features and functions are controlled by software.

- **LLM –** Large Language Models are a type of Generative AI that uses deep learning techniques and massively large data sets to understand, summarize, generate, and predict new content.
- **MAS –** Multi-agent systems are distributed independent intelligent systems that interact and work together to achieve a set of goals.
- **MAPE-K** – Framework for multi-agent systems with capabilities for Monitor, Analyze, Plan, Execute and Knowledge.

### 2.7 RELATIONSHIP WITH OTHER IIC DOCUMENTS

This document references the IIC white paper "Digital Twin Core Conceptual Models and Services"[3]. The characteristics of a digital twin, as defined by Digital Twin Consortium, the IIC's sister consortium, provides the fundamental capability to build adaptive systems.

## 3 MOTIVATION FOR ADAPTIVE SYSTEMS

**Design:** Typically, systems are designed to deliver a pre-defined set of objectives (or goals) while operating in an environment. Designers make certain assumptions about the environment while designing functional and non-functional capabilities of the system. Well-designed systems take due care to address "known knowns" and "known unknowns." This can be represented using the Uncertainty Matrix[4] as shown below. The design is vulnerable to partial information available with the designer (i.e., "unknown knowns") and to the unknowable information (i.e., "unknown unknowns").

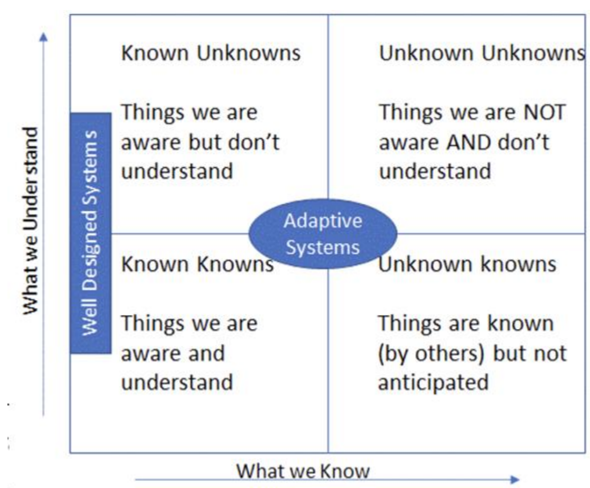


Figure 3-1: Uncertainty matrix.

[3] *https://www.iiconsortium.org/iic-download-digital-twin-core-paper/*
[4] *https://toolbox.nesslabs.com/uncertainty-matrix*

**Software-Aided Systems**: If we look at the examples of mobile phones, Software Driven Networks (SDN) and Software Driven Vehicles (SDV), the content of software in systems has increased steadily. The software in systems and artificial intelligence (AI) have provided flexibility for designers to add features and handle unknown situations discovered ("known unknowns" and to an extent "unknown knowns").

**AI and Generative AI**: Artificial Intelligence combined with Generative AI is a major step towards General Purpose Intelligence. There is a general view that Generative AI is not yet Artificial General Intelligence. This implies that handling "unknown unknowns" is still not a solved solution. Generative AI, with the ability to understand larger problem domains and generate code, has potential to solve the "unknown knowns." Larger language models (LLM) with a vast amount of data can co-relate cross domain knowledge and generate interesting "known" solutions. We will explore the available frameworks in robotics and multi-agent systems to help in solving "unknown knowns" keeping "human-in-middle" towards building adaptive systems.

# 4 Lessons from Industries

In this section we look at how industries like telecom, mobile phone companies and automotive have adopted software driven systems to build products that are adaptive, resilient, enable business agility and provide customer delight.

## 4.1 Telecom Industry – Software Defined Networks

In traditional telecom networks, each device (such as a router or a switch) has its own software and hardware that decides where to send the data packets. This makes the network hard to change and manage, especially when there are many devices and different types of traffic.

In Software Defined Networks (SDNs), the devices are separated into two parts: the data plane and the control plane. The data plane is responsible for forwarding the data packets, while the control plane is responsible for deciding how to forward them. The control plane is centralized in a software application called the SDN controller, which can be implemented with open standards and communicate with all the devices in the network.

By using SDN, telecom operators can create virtual networks that can span across multiple physical locations and devices. They can also dynamically adjust the network configuration and resources according to the demand and traffic patterns of different applications and customers. For example, they can allocate more bandwidth for video streaming or create secure zones for sensitive data.

**Lessons Learned**: By moving to adaptive software driven networks, the telecom operators gained the following:

- Adapt their services and offer new services that require high-speed, low-latency, and reliable connectivity, such as cloud computing, big data analytics, internet of things (IoT), and 5G[5].
- Reduced capital and operational expenses by using cheaper and simpler hardware and by automating network management tasks[6].
- Simplified network administration and management by using centralized software applications and open APIs, which can reduce the need for manual configuration and human intervention, thus saving time and labor costs for network operators[7].
- Enable dynamic and automated provisioning of network services and resources, which can accelerate the delivery of new applications, services, and business models.

Therefore, by moving towards Software Driven Networks, network owners were able to adapt their networks to launch new services, cut costs and optimize network performance.

## 4.2 Mobile Phones

Mobile phones have evolved a lot since they were invented in the 1970s. Here is a brief overview of how mobile phones have changed from being hardware-centric to software-centric devices[8].

The mobile phones launched in 1982 weighed about 10 kilograms[9]. In 1994, the first smart mobile phone IBM Simon Personal Communicator was launched with apps such as address book, calendar, calculator, etc. The launch and impact of iPhone in 2007 is well documented.

Over the years, the focus on improving the hardware components, such as battery, antenna, processor, memory and display played a vital role in bringing the size of the phone down to a shade over 100 grams. Simultaneously, there have been lot of improvements on the software of the smartphones too. The phones' operating systems such as Android or iOS, the apps in the app stores have increased the capability of the phones and provide customer delight.

In 2010, the word 'app' was recognized as word of the year. The software capability of the camera in phones has enabled AI features to be added to the basic camera and has become a differentiating and marketable feature.

---

[5] *https://materialsdatamanagement.com/*

[6] *https://www.networkworld.com/article/963971/what-sdn-is-and-where-its-going.html*

[7] *https://www.javatpoint.com/software-defined-networking-sdn-benefits-and-challenges-of-network-virtualization*

[8] *https://flauntdigital.com/blog/evolution-mobile-phones/*

[9] *https://flauntdigital.com/blog/evolution-mobile-phones/*

**Lessons Learned:** By increasing the software content on the smartphone, the following benefits are achieved:

- Software- and AI-aided features such as smart cameras are marketed as differentiating capabilities of the phone.
- New features are added to phone via software update extending the lifespan of the device.
- End-consumers can customize and adapt the devices to their preferred settings and personalize the look and feel.
- Adoption of new technologies like 5G, AI, and cloud computing to enhance the capabilities and performance of phone providing higher degree of customer delight at a lower cost.

## 4.3 AUTOMOTIVE

Software-defined vehicles (SDV) refers to the state that the vehicle's quantity and value of software (including electronic hardware) exceeds that of the mechanical hardware. Furthermore, it reflects the gradual transformation of automobiles from highly electromechanical terminals to intelligent, expandable mobile electronic terminals that can be continuously upgraded[10]. The following are the lessons learned from the automotive industry.

**Lessons Learned**:

- SDVs give tremendous options for customizing the vehicle with personalized settings and preferences[11]. This increases customer satisfaction.
- Better flexibility and efficiency: Software-defined vehicles can adapt to changing conditions and demands by using software updates. For example, a software update can improve the battery life or the driving range of an electric car[12].
- Higher value and revenue: Software-defined vehicles can increase their value and lifespan by using software updates. For example, a software update can fix a bug or a defect without requiring a physical recall.
- The creation of new business models and revenue streams based on software-defined vehicles, such as offering subscription-based services, pay-per-use features, data monetization, and digital ecosystems.

[10] *https://www2.deloitte.com/cn/en/pages/consumer-business/articles/software-defined-cars-industrial-revolution-on-the-arrow.html*

[11] *https://www.bcg.com/publications/2023/rewriting-rules-of-software-defined-vehicles*

[12] *https://corporate-innovation.co/2022/03/01/the-benefits-and-opportunities-of-software-defined-vehicles/*

Based on these lessons, the information below looks at how software driven industrial systems can be adaptive and provide benefits.

# 5 Generative AI Applied for Adaptive Systems

In this section, we will see examples of using Generative AI for adaptive systems. We will see how researchers are working on leveraging Generative AI for building robots which can be adapted to different tasks. We will also see how multi-agent based adaptive systems can be built.

## 5.1 Generative AI for Adaptive Robotics

With the advent of generative AI, there are lots of activities to provide a natural language interface to robots. However, there are multiple researchers working on using Generative AI to adapt robots that have been developed for a specific purpose. The work is in the following areas:

- A user (potentially non-technical) can sit on the loop, providing high-level feedback through the Generative AI Large Language Model (LLM) while monitoring the robot's performance. These include using prompt structures ("Prompt Engineering") integrated with Robotics OS APIs[13].
- Prompt engineering to generate policy-based codes[14]. This uses the 'Code as Policy' technique to generate code that is secure and governed by policies to adapt behavior of robots. This ensures that the features added to the robot confirm organizational policies and are safe/ secure.
- Prompt engineering to generate optimal plan of execution[15]. This framework generates the steps to execute the robotics activities as plan. Using generative AI to generate the next steps in the plan, it handles diverse situations. Using LLM, summarization, reasoning, and code generation capabilities, provides an innovative approach for robots to handle unplanned situations.
- LLM-driven generation of robot behavioral tree[16]. Behavioral Tree (BT) is an advanced methodology in robotics to specify logic and uses node blocks as code units. The framework demonstrates generation of BT code and the robot's ability to handle unplanned tasks.

[13] *https://www.microsoft.com/en-us/research/uploads/prod/2023/02/ChatGPT___Robotics.pdf*
[14] *https://arxiv.org/abs/2209.07753*
[15] *https://link.springer.com/article/10.1007/s10514-023-10135-3*
[16] *https://arxiv.org/abs/2305.19352*

### 5.2 Generative AI for Multi Agent Systems

Multi-agent system (MAS) is a computerized system composed of multiple interacting intelligent agents. An agent is an entity that can perceive its environment and act upon it to achieve some goals. A MAS can solve problems that are beyond the individual capacities or knowledge of each agent, by coordinating and cooperating with each other. For example, a MAS can be used to model a flock of birds, a team of robots, or a network of smart devices.

A MAPE-K[17] model is a framework for designing self-adaptive multi agent systems, which are systems that can monitor their own behavior and environment and adjust themselves accordingly to achieve desired objectives. MAPE-K stands for Monitor, Analyze, Plan, Execute, and Knowledge, which are the main components of a self-adaptive system:

- The Monitor component collects data from the system and its environment.
- The Analyze component processes the data and detects any changes or anomalies.
- The Plan component generates possible actions or strategies to adapt to the changes. The Execute component implements the chosen actions or strategies.
- The Knowledge component stores and updates the information and models that are shared by the other components[18]. In the Generative AI model, we can introduce LLM in the loop[19]. This enables improved reasoning and decision-making skills.

## 6 Digital Twins – The Path to Self-Adaptive Systems

In this section, we look at how Generative AI integrated with digital twins can enable adaptive systems.

### 6.1 Digital Twins with Generative AI

Digital twins are defined by IIC's sister organization, the Digital Twin Consortium[20], as follows: "A digital twin is a virtual representation of real-world entities and processes, synchronized at a specified frequency and fidelity."

While traditionally digital twins have been built towards bringing together structured information in engineering, production, operations, and enterprise information, few digital twins have integrated with 3D models to improve visualization aspects of the digital twins.

[17] *A Practical Guide to the IBM Autonomic Computing Toolkit*

[18] *https://www.mdpi.com/2076-3417/12/17/8583*

[19] *https://arxiv.org/abs/2307.06187*

[20] *https://www.digitaltwinconsortium.org/initiatives/the-definition-of-a-digital-twin/*

However, a significant portion of industrial knowledge is present as documents and unstructured information. Generative AI contextualizes this information very well and can be used to enrich the knowledge repository of digital twins. Such a digital twin can power many use cases, and use cases relevant for adaptive behavior are elaborated as follows:

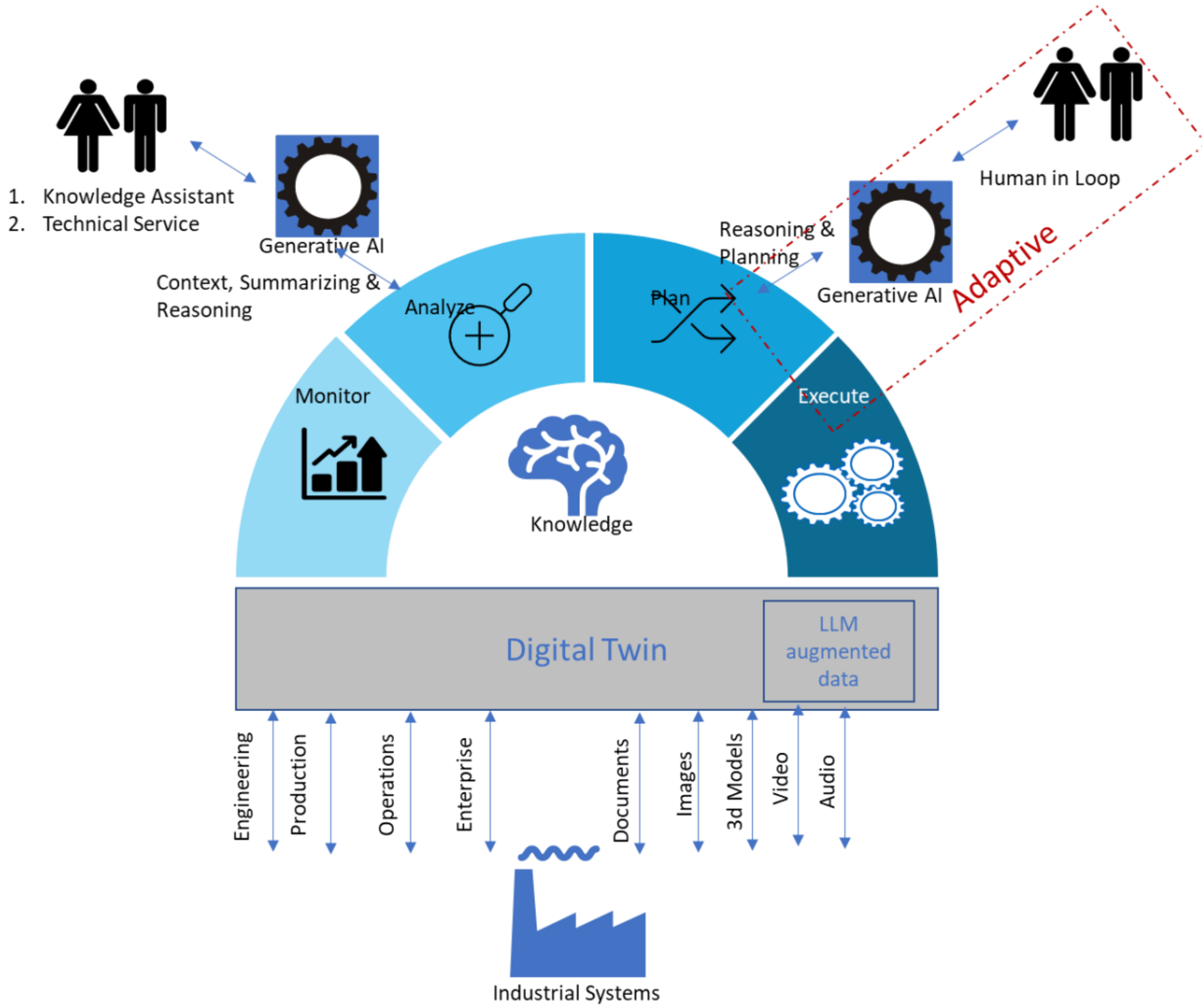


Figure 6-1: Generative AI with digital twin.

## 6.2 Use Cases of Digital Twin with Generative AI

**Knowledge Assistant**: Q&A or Chatbot assistants can be built with GenAI technology to interact with the underlying digital twins for contextual enterprise information. GenAI allows structural information to be fused with unstructured information in a contextual manner. Let's take an example query and review the steps involved:

- Query asked: Give me xx KPI of a manufacturing line.
- LLM interprets the context and invokes the digital twin for xx KPI.
- Digital twin retrieves xx KPI values from operational data store.

- LLM invokes the digital twin for xx KPI specification.
- Digital twin retrieves xx KPI specification from design context store.
- LLM gives response with KPI value and specification to user.

**Technical Service**: Technical service is invoked when a problem is detected either by operator or customer. The technical service agent needs to troubleshoot which might involve looking at the KPI values, identifying root cause of problem and perform resolution tasks. Let's look at the steps involved:

- Problem reported: yy KPI is wrong.
- LLM invokes digital twin for yy API & yy specification documents.
- LLM identifies probable cause and resolution steps from the documents.
- LLM gives the response with root cause and resolution steps.

There are many such use cases possible by integrating GenAI with digital twins for industrial systems. However, we will look at how these two use cases can be extended to build adaptive systems.

## 6.3 ADAPTIVE SYSTEMS WITH GENAI

As shown in the previous use cases, GenAI systems are being used to help in diagnosing problems and answer queries (Refer to Figure 6-1). Leveraging the monitor and analyze modules on top of the digital twin, the modules can now identify whether there is a problem and identify the root cause and possible resolutions.

Based on the Generative Robotics frameworks[21] and the adaptive multi agent frameworks[22] we have the following options to solve "unknown knowns."

- Generate a plan of steps with an option tree for the business user to resolve the problem. Using the information provided by the digital twin, the system can accurately pinpoint the root cause. The resolution steps can be generated and validated by a business user. After validation the steps could be executed by the business user or the system to resolve the problem.
- Generate code to resolve the problem. This requires the system to be able to generate code and integrate into the production code base. This is not a recommended option for now as this will require a mature process to handle automatic code integration.

[21] *https://corporate-innovation.co/2022/03/01/the-benefits-and-opportunities-of-software-defined-vehicles/*
[22] *https://arxiv.org/abs/2305.19352*

- Directly take inputs from business user or IT engineer and act on that. In this option, the business user could directly give instructions using Generative AI to perform steps. This will require the ability of digital twin to take control actions from the Act module.

The frameworks and technologies referred to demonstrate that building adaptive and resilient systems is possible with the technologies available today. Generative AI with the ability to bring context to information, summarize and generate code opens multiple possibilities.

In summary, digital twins provides a platform that enables digital representation of physical systems. This provides the necessary mechanism to infer, understand and control underlying systems in a holistic manner.

Generative AI provides a mechanism to understand “unknown knowns,” identify root causes and generate a plan that could help in resolving the problem. The resolution needs to be reviewed and approved by human-in-middle.

The resolutions could be implemented as generated behavioral tree building blocks so that there is control on what is getting generated and implemented.

## 6.4 Business Benefits

**Resilience**: Adaptive systems can be robust and resilient to failures. They can also adapt well to uncertainties and new scenarios not encountered before.

**Extend life of products/systems**: Adaptive systems extend the life span of products/systems by adapting themselves to new situations. This increase returns from systems and overall return on investment.

**Adding new features**: Adaptive systems with necessary architecture can support the addition of new features by operators, business users, customers and/or IT engineers. This enhances customer satisfaction. It also extends product lifespan and increases return on investment.

## 6.5 Other Considerations

It is easy to get carried away with the generative AI wave and build dynamic adaptive systems. However, one needs to consider the following:

- **Security –** It is important to ensure the human-in-middle is an authenticated and authorized user. A hacker gaining access to such adaptive systems will lead to catastrophic results.
- **Guard rails –** Need to establish boundaries and guard rails around the adaptive systems to ensure that the systems don’t adapt themselves to unintended or malicious purposes.

- **Governance –** Define policies, roles and implement governance to monitor, track behavior of system and authorized users. Review of audit logs, KPIs and purpose of systems should be done in a regular and systemic way.

# 7 SUMMARY

Generative AI is a major step towards achieving full general-purpose AI. Generative AI provides tools which can be leveraged to build autonomic systems. Adaptive systems are a major step towards building autonomic systems. In this article we looked at how Generative AI is used for building adaptive robots and multi-agent systems. We also looked at how industries such as telecom, mobile phones, and automotive adopt software centric design.

By applying the same techniques on industrial digital twins, we can build adaptive and resilient systems. However, as technologies are rapidly evolving, it is recommended to keep human-in-middle to supervise and guide the digital twin to act responsibly. Adaptive systems can extend the lifespan of systems and can create new business models and revenue opportunities.

# 8 REFERENCES

[1] Adaptive behavior of Complex Systems – Embracing Complexity.
*https://hbr.org*

[2] Complex Adaptive Systems.
*https://www.sciencedirect.com/science/article/pii/B9780128037263000079*

[3] Digital Twins for Industrial Applications white paper.
*https://www.iiconsortium.org/digital-twins-for-industrial-applications/*

[4] Thinking Toolbox by Ness Labs.
*https://toolbox.nesslabs.com/uncertainty-matrix*

[5] Materials Data Management - Network | Free Full-Text | Next Generation of SDN in Cloud Fog for 5G and Beyond-Enabled Applications: Opportunities and Challenges.
*https://materialsdatamanagement.com/*

[6] Network World.com – Article 3209131/ What is SDN and Where is it Going?
*https://www.networkworld.com/article/963971/what-sdn-is-and-where-its-going.html*

[7] Javatpoint.com: Software Defined Networking (SDN): Benefits and Challenges of Network Virtualization.
*https://www.javatpoint.com/software-defined-networking-sdn-benefits-and-challenges-of-network-virtualization*

[8], [9] Flaunt Digital.com: The Evolution of Mobile Phones, 1973 – 2019.
*https://flauntdigital.com/blog/evolution-mobile-phones/*

[10] Deloitte.com: Software Defined Vehicles - A Forthcoming Industrial Evolution. *https://www2.deloitte.com/cn/en/pages/consumer-business/articles/software-defined-cars-industrial-revolution-on-the-arrow.html*

[11] Boston Consulting Group: Rewriting the Rules of Software-Defined Vehicles. *https://www.bcg.com/publications/2023/rewriting-rules-of-software-defined-vehicles*

[12] Corporate-Innovation.co: The Benefits and Opportunities of Software-Defined Vehicles – Re-Imagining Corporate Innovation with a Silicon Valley Perspective. *https://corporate-innovation.co/2022/03/01/the-benefits-and-opportunities-of-software-defined-vehicles/*

[13] Micsosoft.com: ChatGPT in middle for Robotics.
*ChatGPT___Robotics.pdf*
*https://www.microsoft.com/en-us/research/uploads/prod/2023/02/ChatGPT___Robotics.pdf*

[14] Policy based code generation using prompt engineering.
*2209.07753.pdf (arxiv.org)*
*https://arxiv.org/abs/2209.07753*

[15] ProgPrompt: program generation for situated robot task planning using large language models | Autonomous Robots (springer.com).
*https://link.springer.com/article/10.1007/s10514-023-10135-3*

[16] [2305.19352] LLM-BRAIn: AI-driven Fast Generation of Robot Behaviour Tree based on Large Language Model (arxiv.org).
*https://arxiv.org/abs/2305.19352*

[17] A Practical Guide to the IBM Autonomic Computing Toolkit - IBM Redbooks - Google Books.
*https://books.google.com/books/about/A_Practical_Guide_to_the_IBM_Autonomic_C.html?id=XHeoSgAACAAJ*

[18] *2307.06187.pdf* Self-Adaptive Large Language Model (LLM)-Based Multiagent Systems *(arxiv.org).*
*https://arxiv.org/abs/2307.06187*

[19] Digital Twins for Industrial Applications white paper.
*https://www.iiconsortium.org/digital-twins-for-industrial-applications/*

[20] Corporate-Innovation.co: The Benefits and Opportunities of Software-Defined Vehicles – Re-Imagining Corporate Innovation with a Silicon Valley Perspective.
*https://corporate-innovation.co/2022/03/01/the-benefits-and-opportunities-of-software-defined-vehicles/*

[21] [2305.19352] LLM-BRAIn: AI-driven Fast Generation of Robot Behaviour Tree based on Large Language Model (arxiv.org),
*https://arxiv.org/abs/2305.19352*

## 9 ACKNOWLEDGEMENTS

The views expressed in the *IIC Journal of Innovation* are the author's views and do not necessarily represent the views of their respective employers nor those of the Industry IoT Consortium®.



- Return to *IIC Journal of Innovation landing page* for more articles and past editions.